



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/750,002

12/30/2003

Hans-Christoph Rohland

6570P009

9951

45062

7590

02/05/2009

SAP/BSTZ

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP

1279 OAKMEAD PARKWAY

SUNNYVALE, CA 94085-4040

EXAMINER

ALI, FARHAD

ART UNIT

PAPER NUMBER

2446

MAIL DATE

DELIVERY MODE

02/05/2009

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/750,002	<b>Applicant(s)</b> ROHLAND ET AL.	
	<b>Examiner</b> FARHAD ALI	<b>Art Unit</b> 2446	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 09 June 2008.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-25 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-25 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)            | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)   | Paper No(s)/Mail Date. _____                                      |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date <u>10/13/2008 and 01/02/2009</u> .                               | 6) <input type="checkbox"/> Other: _____                          |

## **DETAILED ACTION**

### **Status of Claims:**

**Claims 1-25 are pending in this Office Action.**

**Claims 6 and 18 are amended.**

### ***Claim Rejections - 35 USC § 103***

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 1-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zargham et al. (US 6,954,757 B2) in view of Vincent et al. (US 7,197,533 B2).

### **Claim 1**

Zargham teaches a system comprising:

a database ([Zargham] Column 8 Lines 34-37, "Central repository--refers to a sharable unified capacity such as the operational data store (ODS) with a relational database management system (RDBMS) in the ZLE framework"); and

a plurality of instances of an application server implementing a Java application model ([Zargham] Column 16 Lines 12-16, "The workflow service in the ZLE framework is, for example, an EJB (Enterprise Java Bean, Java 2 enterprise edition ( J2EE)) compliant service running on parallel, available application servers that can store its

Art Unit: 2446

workflow as XML data structures”) coupled in a star topology with the message server at a center of the star topology, the plurality of instances sharing the database ([Zargham] Column 3 Lines 21-26, “the ZLE framework defines a multilevel architecture with a hub, wherein the enterprise applications are loosely coupled to the hub and communicating therewith via adapters” and see Column 6 Lines 53-61, “Loosely coupled applications”).

Zarhgham does not disclose a message server having no persistent state.

Vincent teaches support for a non-persistent service in Column 5 Lines 9-19, “FIG. 3A is a flowchart depicting the operation and control flow of a conventional process for handling transactions from a non-persistent service. Specifically, the flowchart of FIG. 3A shows a conventional process for providing support to a non-persistent service, when no error occurs during the execution of the application code and the transaction is fully completed. The flowchart of FIG. 3A should be viewed with reference to FIGS. 1 and 2, which describe in more detail the overall architecture of the system. In FIG. 3A, the exemplary non-persistent service depicted is an instant messaging service, as served by a middleware server 106” in order to “provide a web application server that provides support for a non-persistent service” (Column 2 lines 1-3).

It would have been obvious to one of ordinary skill in the art at the time to create the invention of Zargham to include the non-persistent service as taught by Vincent in order to “provide a web application server that provides support for a non-persistent service” (Column 2 lines 1-3).

**Claim 2**

The modified Zargham teaches the system of claim 1 wherein each instance comprises:

a dispatcher node; and a plurality of server nodes ([Zargham] See Figure 9, ZLE framework).

**Claim 3**

The modified Zargham teaches the system of claim 2 wherein each server node comprises:

a java 2 enterprise edition ( J2EE) engine ([Zargham] Column 16 Lines 12-16, “The workflow service in the ZLE framework is, for example, an EJB (Enterprise Java Bean, Java 2 enterprise edition ( J2EE)) compliant service running on parallel, available application servers that can store its workflow as XML data structures”).

**Claim 4**

The modified Zargham teaches the system of claim 1 further comprising:

a central lock server to provide cluster wide locks to the plurality of instances ([Zargham] Column 7 Lines 43-46, “An event may unlock or prompt the commencement of one or more business transactions. An event may lock or prompt the ending of one or more business transactions”).

**Claim 5**

The modified Zargham teaches the system of claim 1 wherein the message server comprises:

a first data structure to store a list of connected clients; and a second data structure and a list of services provided in the system ([Zargham] Column 1 Lines 43-46, “the ZLE can integrate data related to the real time operations of the enterprise into a data storage cache, also known as operational data store (ODS)”).

**Claim 6**

Zargham teaches a computer readable storage media containing executable computer program instructions which when executed cause a digital processing system to perform a method comprising:

starting a central services node to provide a locking service and a messaging service ([Zargham] Column 7 Lines 43-46, “An event may unlock or prompt the commencement of one or more business transactions. An event may lock or prompt the ending of one or more business transactions” and Column 6 Lines 62-67, “Tightly coupled applications—refers to applications that are not stand-alone and are tightly integrated into the ZLE framework. Tightly integrated functionality—e.g., event capture, data extraction, rules, workflow, message transports and transformations—becomes part of the ZLE core functionality”);

starting a plurality of application server instances ([Zargham] Column 16 Lines 12-16, “The workflow service in the ZLE framework is, for example, an EJB (Enterprise

Art Unit: 2446

Java Bean, Java 2 enterprise edition ( J2EE)) compliant service running on parallel, available application servers that can store its workflow as XML data structures”); and

organizing the application server instances into a cluster having star topology with the central services node at a center of the star topology ([Zargham] Column 3 Lines 21-26, “the ZLE framework defines a multilevel architecture with a hub, wherein the enterprise applications are loosely coupled to the hub and communicating therewith via adapters” and see Column 6 Lines 53-61, “Loosely coupled applications”).

Zargham does not disclose the messaging service having no persistent state.

Vincent teaches support for a non-persistent service in Column 5 Lines 9-19, “FIG. 3A is a flowchart depicting the operation and control flow of a conventional process for handling transactions from a non-persistent service. Specifically, the flowchart of FIG. 3A shows a conventional process for providing support to a non-persistent service, when no error occurs during the execution of the application code and the transaction is fully completed. The flowchart of FIG. 3A should be viewed with reference to FIGS. 1 and 2, which describe in more detail the overall architecture of the system. In FIG. 3A, the exemplary non-persistent service depicted is an instant messaging service, as served by a middleware server 106” in order to “provide a web application server that provides support for a non-persistent service” (Column 2 lines 1-3).

It would have been obvious to one of ordinary skill in the art at the time to create the invention of Zargham to include the non-persistent service as taught by Vincent in

Art Unit: 2446

order to "provide a web application server that provides support for a non-persistent service" (Column 2 lines 1-3).

### **Claim 7**

The modified Zargham teaches the computer readable storage media of claim 6 containing executable computer program instructions which when executed cause a digital processing system to perform the method further comprising:

sharing a database among the plurality of application server instances ([Zargham] Column 8 Lines 34-37, "Central repository--refers to a sharable unified capacity such as the operational data store (ODS) with a relational database management system (RDBMS) in the ZLE framework").

### **Claim 8**

The modified Zargham teaches the computer readable storage media of 6 containing executable computer program instructions which when executed cause a digital processing system to perform the method wherein starting a plurality of application server instances comprises:

starting, for each application server instance of the plurality, a dispatcher node and a plurality of server nodes ([Zargham] See Figure 9, ZLE framework).



**Claim 9**

Zarhgham does not disclose: starting a message server having no persistent state.

Vincent teaches support for a non-persistent service in Column 5 Lines 9-19, "FIG. 3A is a flowchart depicting the operation and control flow of a conventional process for handling transactions from a non-persistent service. Specifically, the flowchart of FIG. 3A shows a conventional process for providing support to a non-persistent service, when no error occurs during the execution of the application code and the transaction is fully completed. The flowchart of FIG. 3A should be viewed with reference to FIGS. 1 and 2, which describe in more detail the overall architecture of the system. In FIG. 3A, the exemplary non-persistent service depicted is an instant messaging service, as served by a middleware server 106" in order to "provide a web application server that provides support for a non-persistent service" (Column 2 lines 1-3).

It would have been obvious to one of ordinary skill in the art at the time to create the invention of Zargham to include the non-persistent service as taught by Vincent in order to "provide a web application server that provides support for a non-persistent service" (Column 2 lines 1-3).

**Claim 10**

The modified Zargham teaches the computer readable storage media of claim 6 containing executable computer program instructions which when executed cause a digital processing system to perform the method further comprising:

registering each application server with the messaging server ([Zargham] Column 13 Lines 66-68, "The robust message store function supports the EAI platform for ZLE hub-based publish and subscribe operations").

**Claim 11**

The modified Zargham teaches the computer readable storage media of claim 6 containing executable computer program instructions which when executed cause a digital processing system to perform the method further comprising:

conducting inter instance communication through the messaging service ([Zargham] Column 21 Lines 53-55, "Messaging functions in the ZLE framework may involve a simple messaging scenario of an EAI-type request-response situation").

**Claim 12**

The modified Zargham teaches the computer readable storage media of claim 9 containing executable computer program instructions which when executed cause a digital processing system to perform the method further comprising:

restarting the message server without state recovery responsive to a system failure ([Zargham] Column 18 Lines 46-51, "This also means the ability to monitor

Art Unit: 2446

transactions (such as the above-mentioned business transactions) and restart them in the event of failure, manage transaction boundaries, manage queues, and so on” and a memory server having no persistent state typically cannot have a state recovery).

### **Claim 13**

The modified Zargham teaches the computer readable storage media of claim 10 containing executable computer program instructions which when executed cause a digital processing system to perform the method further comprising:

notifying all registered instances from the message server when an additional instance joins the cluster ([Zargham] Column 13 Lines 66-68, “The robust message store function supports the EAI platform for ZLE hub-based publish and subscribe operations”).

### **Claim 14**

Zargham teaches a system comprising:

means for organizing a plurality of application servers instances into a cluster having a star topology with a central services node at a center of the star topology ([Zargham] Column 3 Lines 21-26, “the ZLE framework defines a multilevel architecture with a hub, wherein the enterprise applications are loosely coupled to the hub and communicating therewith via adapters” and see Column 6 Lines 53-61, “Loosely coupled applications”);

Art Unit: 2446

means for sharing a storage resource across the cluster; and means for performing centralized inter instances communication ([Zargham] Column 8 Lines 34-37, "Central repository--refers to a sharable unified capacity such as the operational data store (ODS) with a relational database management system (RDBMS) in the ZLE framework").

Zargham et al. does not disclose means for performing centralized inter instances communication without maintenance of persistent state information

Vincent teaches support for a non-persistent service in Column 5 Lines 9-19, "FIG. 3A is a flowchart depicting the operation and control flow of a conventional process for handling transactions from a non-persistent service. Specifically, the flowchart of FIG. 3A shows a conventional process for providing support to a non-persistent service, when no error occurs during the execution of the application code and the transaction is fully completed. The flowchart of FIG. 3A should be viewed with reference to FIGS. 1 and 2, which describe in more detail the overall architecture of the system. In FIG. 3A, the exemplary non-persistent service depicted is an instant messaging service, as served by a middleware server 106" in order to "provide a web application server that provides support for a non-persistent service" (Column 2 lines 1-3).

It would have been obvious to one of ordinary skill in the art at the time to create the invention of Zargham to include the non-persistent service as taught by Vincent in order to "provide a web application server that provides support for a non-persistent service" (Column 2 lines 1-3).

**Claim 15**

The modified Zargham teaches the system of claim 14 further comprising:  
means for centralized locking of a resource within the cluster ([Zargham] Column 7 Lines 43-46, "An event may unlock or prompt the commencement of one or more business transactions. An event may lock or prompt the ending of one or more business transactions").

**Claim 16**

Zargham et al. does not disclose: a message server having no persistent state.  
Vincent teaches support for a non-persistent service in Column 5 Lines 9-19, "FIG. 3A is a flowchart depicting the operation and control flow of a conventional process for handling transactions from a non-persistent service. Specifically, the flowchart of FIG. 3A shows a conventional process for providing support to a non-persistent service, when no error occurs during the execution of the application code and the transaction is fully completed. The flowchart of FIG. 3A should be viewed with reference to FIGS. 1 and 2, which describe in more detail the overall architecture of the system. In FIG. 3A, the exemplary non-persistent service depicted is an instant messaging service, as served by a middleware server 106" in order to "provide a web application server that provides support for a non-persistent service" (Column 2 lines 1-3).

Art Unit: 2446

It would have been obvious to one of ordinary skill in the art at the time to create the invention of Zargham to include the non-persistent service as taught by Vincent in order to “provide a web application server that provides support for a non-persistent service” (Column 2 lines 1-3).

### **Claim 17**

The modified Zargham teaches the system of claim 14 wherein the means for performing comprises:

means for registering instances; and

means for recording services provided in the cluster ([Zargham] Column 13 Lines 66-68, “The robust message store function supports the EAI platform for ZLE hub-based publish and subscribe operations”).

### **Claim 18**

Zargham teaches a method comprising:

starting a central services node to provide a locking service and a messaging service ([Zargham] Column 7 Lines 43-46, “An event may unlock or prompt the commencement of one or more business transactions. An event may lock or prompt the ending of one or more business transactions” and Column 6 Lines 62-67, “Tightly coupled applications—refers to applications that are not stand-alone and are tightly integrated into the ZLE framework. Tightly integrated functionality—e.g., event capture,

Art Unit: 2446

data extraction, rules, workflow, message transports and transformations—becomes part of the ZLE core functionality”).

starting a plurality of application server instances ([Zargham] Column 16 Lines 12-16, “The workflow service in the ZLE framework is, for example, an EJB (Enterprise Java Bean, Java 2 enterprise edition ( J2EE)) compliant service running on parallel, available application servers that can store its workflow as XML data structures”); and

organizing the application server instances into a cluster having star topology with the central services node at a center of the star topology ([Zargham] Column 3 Lines 21-26, “the ZLE framework defines a multilevel architecture with a hub, wherein the enterprise applications are loosely coupled to the hub and communicating therewith via adapters” and see Column 6 Lines 53-61, “Loosely coupled applications”).

Zarhgham does not disclose the messaging service not maintaining a persistent state.

Vincent teaches support for a non-persistent service in Column 5 Lines 9-19, “FIG. 3A is a flowchart depicting the operation and control flow of a conventional process for handling transactions from a non-persistent service. Specifically, the flowchart of FIG. 3A shows a conventional process for providing support to a non-persistent service, when no error occurs during the execution of the application code and the transaction is fully completed. The flowchart of FIG. 3A should be viewed with reference to FIGS. 1 and 2, which describe in more detail the overall architecture of the system. In FIG. 3A, the exemplary non-persistent service depicted is an instant messaging service, as served by a middleware server 106” in order to “provide a web

Art Unit: 2446

application server that provides support for a non-persistent service" (Column 2 lines 1-3).

It would have been obvious to one of ordinary skill in the art at the time to create the invention of Zargham to include the non-persistent service as taught by Vincent in order to "provide a web application server that provides support for a non-persistent service" (Column 2 lines 1-3).

#### **Claim 19**

The modified Zargham teaches the method of claim 18 further comprising:  
sharing a database among the plurality of application server instances  
([Zargham] Column 8 Lines 34-37, "Central repository--refers to a sharable unified capacity such as the operational data store (ODS) with a relational database management system (RDBMS) in the ZLE framework").

#### **Claim 20**

The modified Zargham teaches the method of claim 18 wherein starting a plurality of application server instances comprises:  
starting, for each instance of the plurality, a dispatcher node and a plurality of server nodes ([Zargham] See Figure 9, ZLE framework).



**Claim 21**

Zarhgham et al. does not disclose starting a message server having no persistent state.

Vincent teaches support for a non-persistent service in Column 5 Lines 9-19, "FIG. 3A is a flowchart depicting the operation and control flow of a conventional process for handling transactions from a non-persistent service. Specifically, the flowchart of FIG. 3A shows a conventional process for providing support to a non-persistent service, when no error occurs during the execution of the application code and the transaction is fully completed. The flowchart of FIG. 3A should be viewed with reference to FIGS. 1 and 2, which describe in more detail the overall architecture of the system. In FIG. 3A, the exemplary non-persistent service depicted is an instant messaging service, as served by a middleware server 106" in order to "provide a web application server that provides support for a non-persistent service" (Column 2 lines 1-3).

It would have been obvious to one of ordinary skill in the art at the time to create the invention of Zargham to include the non-persistent service as taught by Vincent in order to "provide a web application server that provides support for a non-persistent service" (Column 2 lines 1-3).

**Claim 22**

The modified Zargham teaches the method of claim 18 wherein organizing comprises:

Art Unit: 2446

registering each application server with the messaging server ([Zargham] Column 13 Lines 66-68, "The robust message store function supports the EAI platform for ZLE hub-based publish and subscribe operations").

#### **Claim 23**

The modified Zargham teaches the method of claim 18 further comprising:  
conducting inter instance communication through the messaging service ([Zargham] Column 21 Lines 53-55, "Messaging functions in the ZLE framework may involve a simple messaging scenario of an EAI-type request-response situation").

#### **Claim 24**

The modified Zargham teaches the method of claim 21 further comprising:  
restarting the message server without state recovery responsive to a system failure ([Zargham] Column 18 Lines 46-51, "This also means the ability to monitor transactions (such as the above-mentioned business transactions) and restart them in the event of failure, manage transaction boundaries, manage queues, and so on" and a memory server having no persistent state typically cannot have a state recovery).

#### **Claim 25**

The modified Zargham teaches the method of claim 22 wherein organizing further comprises:

Art Unit: 2446

notifying all registered instances from the message server when an additional instance joins the cluster ([Zargham] Column 13 Lines 66-68,"The robust message store function supports the EAI platform for ZLE hub-based publish and subscribe operations").

### ***Response to Arguments***

3. Applicant's arguments with respect to independent claims have been considered but are moot in view of the new ground(s) of rejection.

### ***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to FARHAD ALI whose telephone number is (571)270-1920. The examiner can normally be reached on Monday thru Friday, 7:30am to 5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jeffrey C. Pwu can be reached on (571) 272-6798. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2446

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Farhad Ali/  
Examiner, Art Unit 2446

/Jeffrey Pwu/  
Supervisory Patent Examiner, Art Unit 2446